下载APP

17 | 图像分类（上）：图像分类原理与图像分类模型

2021-11-19 方远

《PyTorch深度学习实战》 课程介绍 >

讲述：方远
时长 15:56 大小 14.61M

你好，我是方远，欢迎来到图像分类的学习。

通过前面的学习，我们已经掌握了 PyTorch 有关深度学习的不少知识。为了避免纸上谈兵，我们正式进入实战环节，分别从计算机视觉与自然语言处理这两个落地项目最多的深度学习应用展开，看看业界那些常见深度学习应用都是如何实现的。

领资料

完成这个模块的学习以后，我想你不仅仅会巩固之前学习的内容，还会进一步地落实到细分的领域去看待问题、解决问题。

说到计算机视觉，**很常见的一种应用方向就是图像分类**。关于图像分类，其实离我们并不遥远。你有没有发现，现在很多智能手机，照相的时候都会自动给照片内容打上标签。

举个例子，你看后面的截图，就是我用手机拍照的时候，手机自动对摄像头的内容进行了识别，打上了“多云”这个标签。

然后你会发现，手机还能根据识别到的内容，为你推荐一些美化的方案。那这是怎么做到的呢？其实这就是卷积神经网络最常用、最广泛且最基本的一个应用：图像分类。

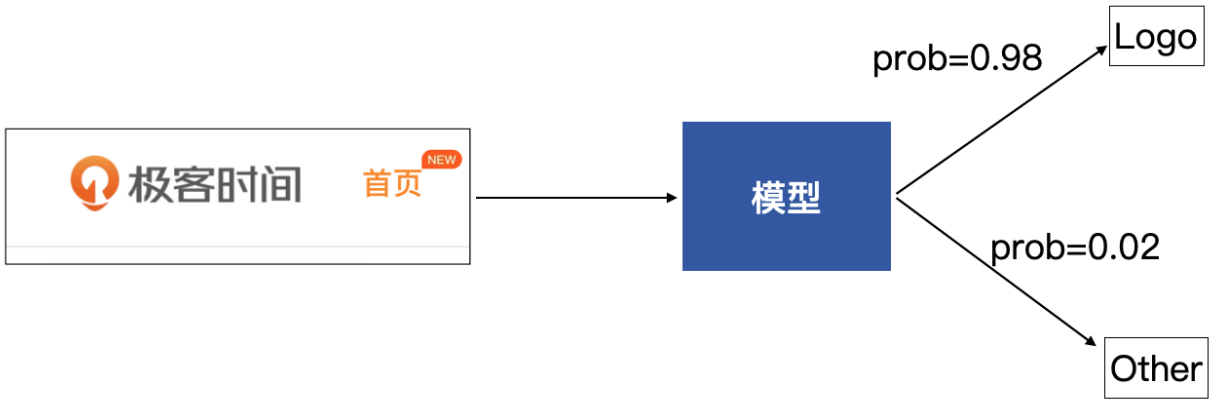
今天咱们就来一探究竟，看看图像分类到底是怎么一回事。我会用两节课的篇幅，带你学习图像分类。这节课我们先学习理论知识，掌握图像分类原理和常见的卷积神经网络。下节课，我们再基于今天学到的原理，一块完成一个完整的图像分类项目实践。

图像分类原理

我们还是“书接上文”，沿用第 3 节课 NumPy 的那个例子。现在线上每天都有大量的图片被上传，老板交代你设计一个模型，把有关极客时间 Logo 的图片自动找出来。

把这个需求翻译一下就是：建立一个图像分类模型，提供自动识别有极客时间 Logo 图片的功能。

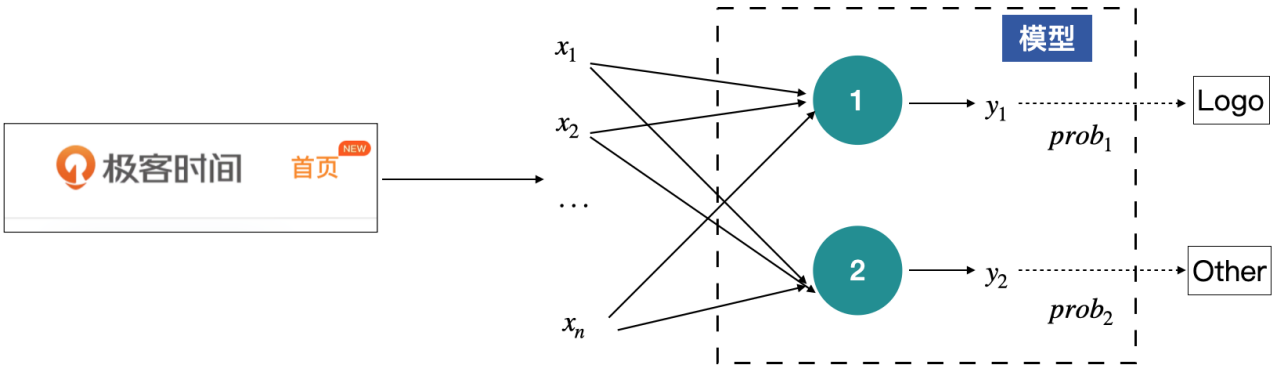
我们来梳理一下这个模型的功能，我们这个模型会接收一张图片，然后会输出一组概率，分别是该图片为 Logo 的概率与该图片为其他图片的概率，从而通过概率来判断这张图片是 Logo 类还是 Other 类，如下图所示：



感知机

我们将上面的模型进一步拆分，看看如何才能获得这样的一组输出。

其中输入的图片，就是输入 X，将其展开后，可以获得输入 X 为 $X = x_1, x_2, \ldots, x_n$，而模型可以看做有两个节点，每个节点都会有一个输出，分别代表着对输入为 Logo 和 Other 的判断，但这里的输出暂时还不是概率，只是模型输出的一组数值。这一部分内容如下图所示：



上图这个结构其实就是感知机了，中间绿色的节点叫做神经元，是感知机的最基本组成单元。上图中的感知机只有中间一层（绿色的神经元），如果有多层神经元的话，我们就称之为多层感知机。

那什么是神经元呢？神经元是关于输入的一个线性变换，每一个输入 x 都会有一个对应的权值，上图中的 y 的计算方式为：

$$y_i = \delta(w_{i1}x_1 + w_{i2}x_2 + \ldots + w_{i_n}x_n + b_i), \quad i = 1, 2$$

其中，$w_{i1}, w_{i2}, \ldots, w_{in}$ 是神经元的权重，b_i 为神经元的偏移项。权重与偏移项都是通过模型学习到的参数。δ 为激活函数，激活函数是一个可选参数。

那如何将一组数值，也就是 y_1 与 y_2 转换为一组对应的概率呢？这个时候 Softmax 函数就要登场了。它的作用就是将一组数值转换为对应的概率，概率和为 1。

Softmax 的计算公式如下：

$$\delta(x_j) = \frac{e^{x_j}}{\sum_{j=1}^{m} e^{x_j}}$$

请看下面的代码，我们用 Softmax 函数对原始的输入 y 做个转化，将 y 中的数值转化为一组对应的概率：

复制代码

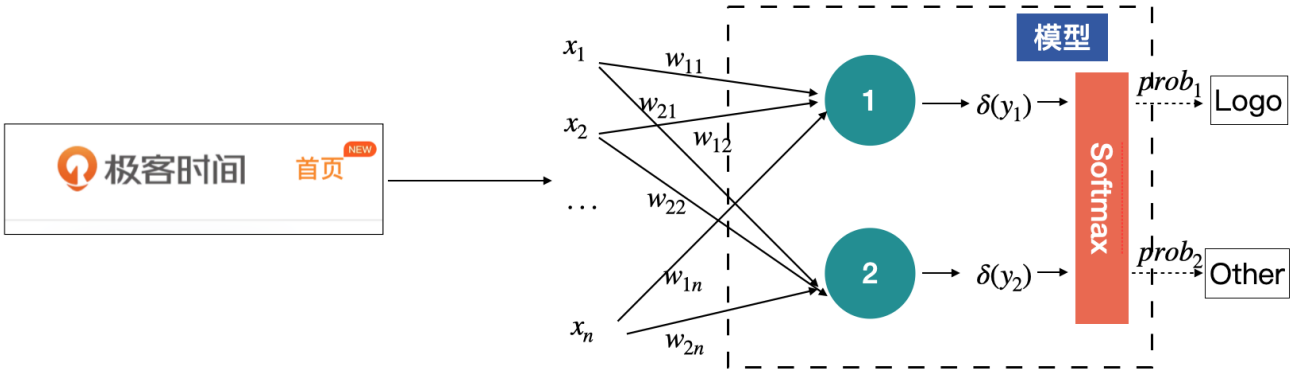
```
import torch
import torch.nn as nn

# 2个神经元的输出y的数值为
y = torch.randn(2)
print(y)
输出：tensor([0.2370, 1.7276])
m = nn.Softmax(dim=0)
out = m(y)
print(out)
输出：tensor([0.1838, 0.8162])
```

你看，经过 Softmax 之后，原始的输出 y 是不是转换成一组概率，并且概率的和为 1 呢。原始 y 中最大的 y 具有最大的概率。

当然，Softmax 也不是每一个问题都会使用。我们根据问题的不同可以采用不同的函数，例如，有的时候也会使用 sigmoid 激活函数，sigmoid 激活函数是将 1 个数值转换为 0 到 1 之间的概率。

现在，我们将上述的过程补充到前面的模型里，如下图所示。



全连接层

其实，上面那张示意图，就是图像的分类原理了。其中绿色那一层。在卷积神经网络中称为**全连接层，Full Connection Layer，简称 fc 层。一般都是放在网络的最后端**，用来获得最终的输出，也就是各个类别的概率。

因为全连接层中的神经元的个数是固定的，所以说在有全连接层的网络中，输入图片是必须固定尺寸的。而现实里我们线上收集到的图片会有不同的尺寸，所以需要先把图片尺寸统一起来，PyTorch 才能进一步处理。

我们假设将前面的输入图片 resize 到 128x128，然后看看全连接层推断的过程在 PyTorch 中是如何实现的。

复制代码

```
x = torch.randint(0, 255, (1, 128*128), dtype=torch.float32)
fc = nn.Linear(128*128, 2)
y = fc(x)
print(y)
输出：tensor([[  72.1361, -120.3565]], grad_fn=<AddmmBackward>)
# 注意y的shape是(1, 2)
output = nn.Softmax(dim=1)(y)
print(output)
输出：tensor([[1., 0.]], grad_fn=<SoftmaxBackward>)
```

结合代码不难看出，PyTorch 中全连接层用 nn.Linear 来实现。我们分别看看里面的重要参数有哪些：

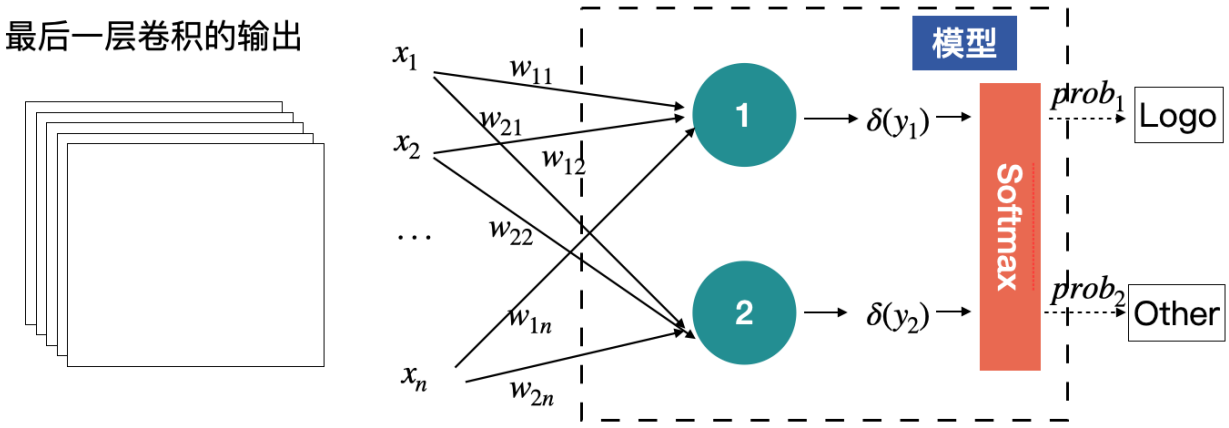
in_features：输入特征的个数，在本例中为 128x128；

out_features：输出的特征数，在本例中为 2；

bias：是否需要偏移项，默认为 True。

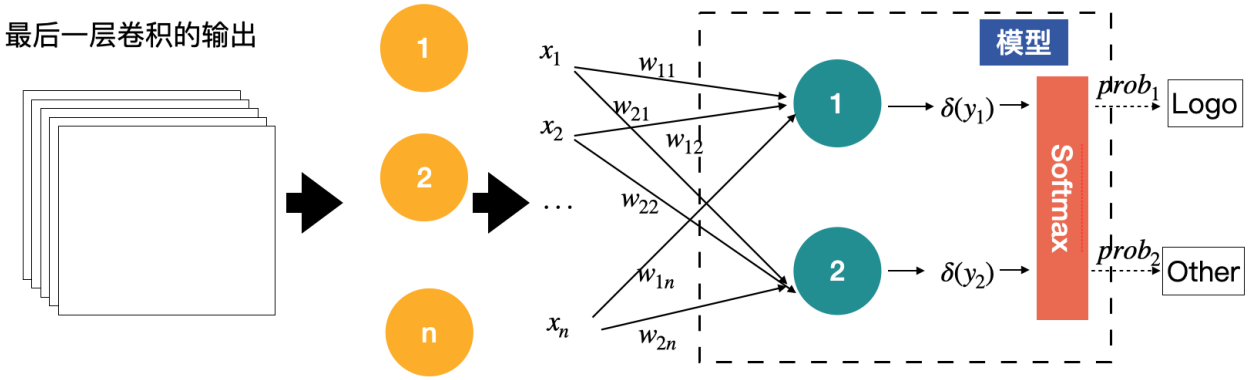
全连接层的输入，也不是原始图片数据，而是经过多层卷积提取的特征。

前面我们曾说过，有的网络是可以接收任意尺度的输入的。在上文中的设计中，全连接层的输入 x1 到 xn 是固定的，数目等于最后一层特征图所有元素的数目。如下图所示：



我们将上述结构稍作调整，就可以接收任意尺度的输入了。只需要在最后的特征图后面加一个全局平均即可，也就是将每个特征图进行求平均，用平均值代替特征图，这样无论输入的尺度是多少，进入全连接层的数据量都是固定的。

如下图所示，黄色的圈就是全局平均的结果。

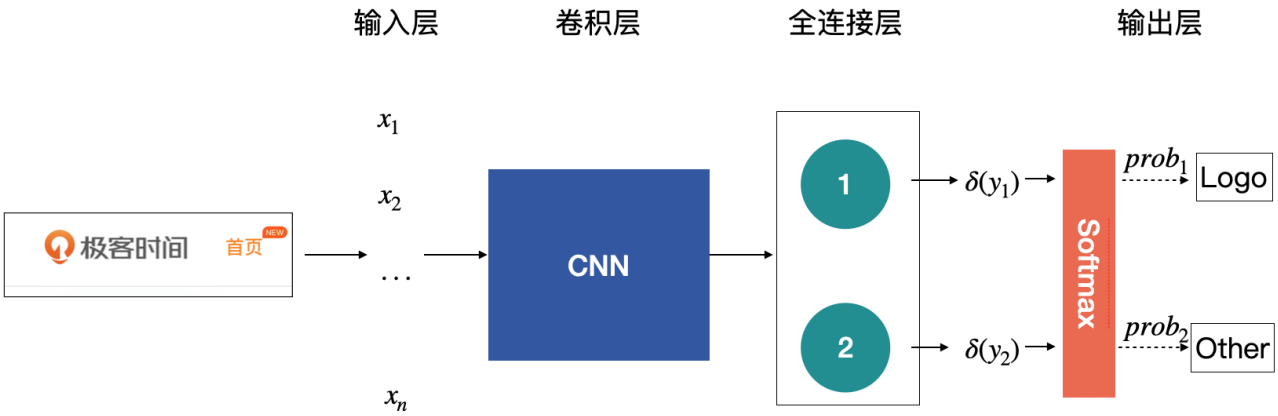


我们下一节课介绍的 EfficientNet 就是采用这种方式，使得网络可以使用任意尺度的图片进行训练。

卷积神经网络

其实刚才说的多层感知机就是卷积神经网络的前身，由于自身的缺陷（参数量大、难以训练），使其在历史上有段时间一直是停滞不前，直到卷积神经网络的出现，打破了僵局。

卷积神经网络的最大作用就是提取出输入图片的丰富信息，然后再对接上层的一些应用，比如前面提到的图片分类。把卷积神经网络应用到图像分类原理中，得到的模型如下图所示：



你需要注意的是示意图中各个层的定义，不同层有不同的名称。

在上图中，**整个模型或者网络的重点全都在卷积神经网络那块，所以这也是我们的工作重点。**

那如何找到一个合适的卷积神经网络呢？在实际工作中，我们几乎不会自己去设计一个神经网络网的（因为不可控的变量太多），而是直接选择一些大神设计好的网络直接使用。那网络模型那么多，我们如何验证大神们提出的网络确实是可靠、可用的呢？

ImageNet

在业界中有个标杆——ImageNet，大家都用它来评价提出模型的好与坏。

ImageNet 本身包含了一个非常大的数据集，并且从 2010 年开始，每年都会举办一次著名的 ImageNet 大规模视觉识别挑战赛（The ImageNet Large Scale Visual Recognition Challenge ，ILSVRC），比赛包含了图像分类、目标检测与图像分割等任务。

其中，图像分类比赛使用的数据集是一份有 1000 个类别的庞大数据集，只要能在这个比赛中脱颖而出的模型，都是我们所说的经典网络结构，这些网络在实际项目中基本都是我们的首选。

从 2012 年开始，伴随着深度学习的发展，几乎每一年都有非常经典的网络结构诞生，下表为历年来 ImageNet 上 Top-5 的错误率。

竞赛年份	Top5错误率
2010	28.20%
2011	25.80%
2012(AlexNet)	16.40%
2014(VGG)	7.30%
2014(GoogLeNet)	6.70%
2015(ResNet)	3.57%
2017(SENet)	2.30%
Human	5.00%

极客时间

你可能会有疑问，了解这么多网络模型真的有必要么？

我想说的是，磨刀不误砍柴工，**机器学习这个领域始终是依靠研究驱动的。**工作当中，我们很少从 0 到 1 自创一个网络模型，常常是在经典设计基础上做一些自定义配置，所以你最好对这些经典网络都有所了解。

接下来，我们就挑选几个经典的神经网络来看看。

VGG

VGG取得了 ILSVRC 2014 比赛分类项目的第 2 名和定位项目的第 1 名的优异成绩。

当年的 VGG 一共提供了 A 到 E6 种不同的 VGG 网络（字母不同，只是表示层数不一样）。VGG19 的效果虽说最好，但是综合模型大小等指标，在实际项目中 VGG16 用得更加多一点。具体的网络结构你可以看看论文。

我们来看看 VGG 突破的一些重点：

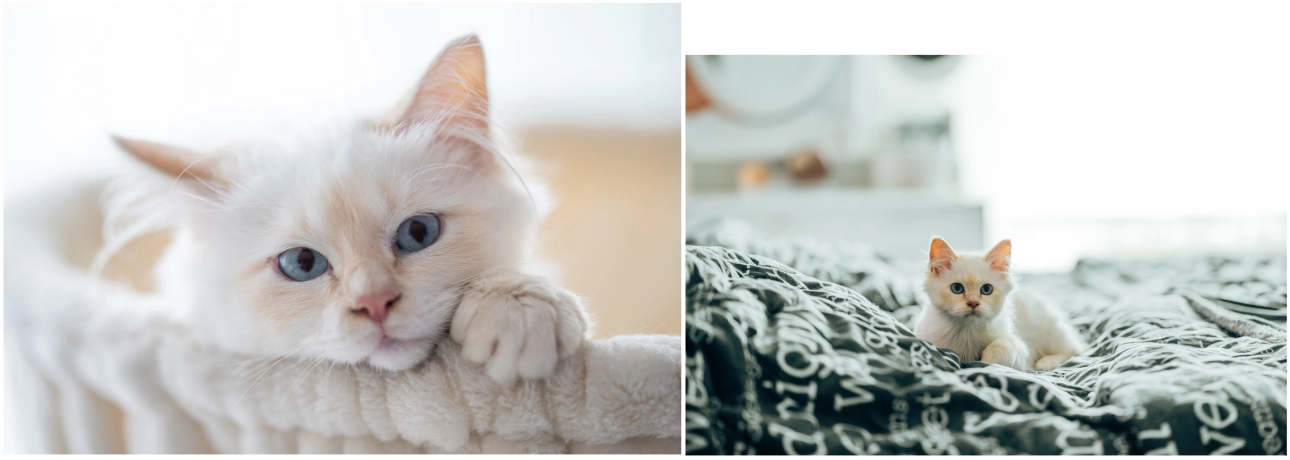
1. 证明了随着模型深度的增加，模型效果也会越来越好。
2. 使用较小的 3x3 的卷积，代替了 AlexNet 中的 11x11、7x7 以及 5x5 的大卷积核。

关于第二点，VGG 中将 5x5 的卷积用 2 层 3x3 的卷积替换；将 7x7 的卷积用 3 层 3x3 的卷积替换。这样做首先可以减少网络的参数，其次是可以在相同感受野的前提下，加深网络的层数，从而提取出更加多样的非线性信息。

GoogLeNet

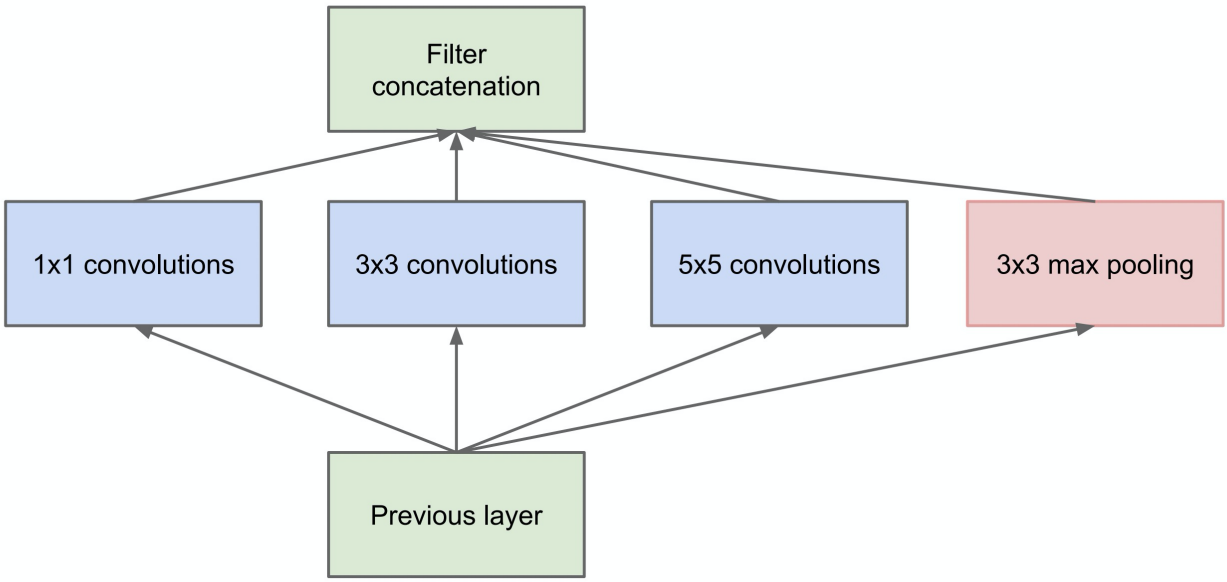
2014 年分类比赛的冠军是GoogLeNet（VGG 同年）。GoogLeNet 的核心是 Inception 模块。这个时期的 Inception 模块是 v1 版本，后续还有 v2、v3 以及 v4 版本。

我们先来看看 GoogLeNet 解决了什么样的问题。研究人员发现，对于同一个类别的图片，主要物体在不同图片中，所占的区域大小均有不同，例如下图所示。



如果使用 AlexNet 或者 VGG 中标准的卷积的话，每一层只能以相同的尺寸的卷积核来提取图片中的特征。

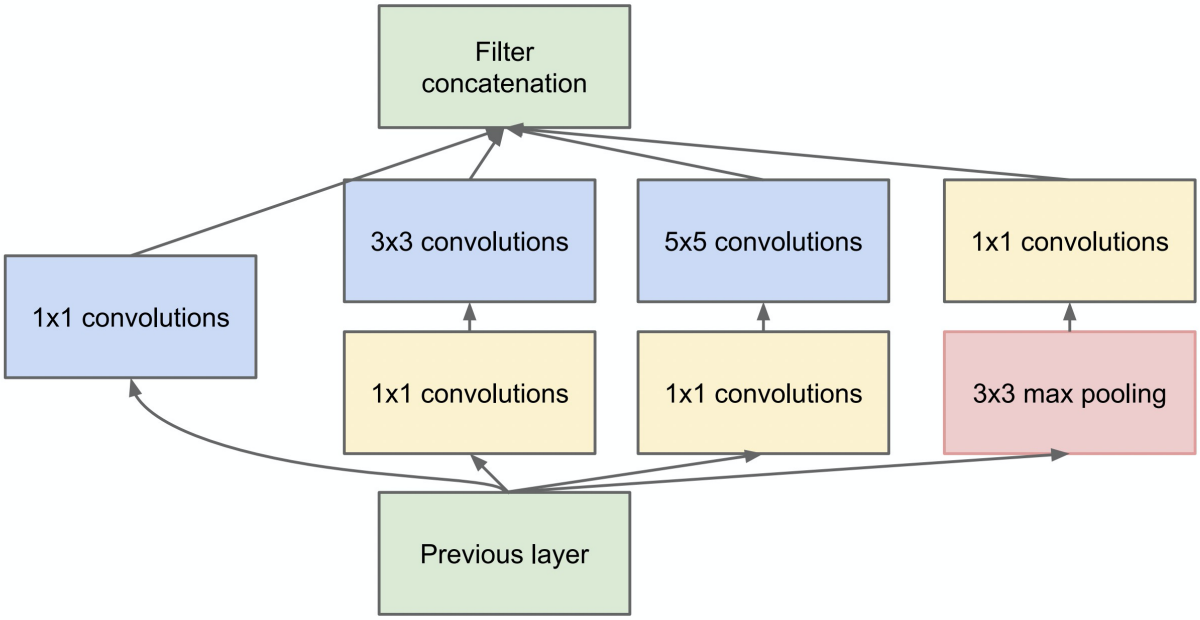
但是正如上图所示，很可能物体以不同的尺寸出现在图片中，那么能否以不同尺度的卷积来提取不同的特征呢？沿着这个想法，Inception 模块应运而生，如下图示：



图片来源：https://arxiv.org/abs/1409.4842

结合图示我们发现，这里是将原来的相同尺寸卷积提取特征的方式拆分为，使用 1x1、3x3、5x5 以及 3x3 的 max pooling 同时进行特征提取，然后再合并到一起。这样就做到了以**多尺度的方式**提取图片中的特征。

作者为了降低网络的计算成本，将上述的 Inception 模块做了一步改进，在 3x3、5x5 之前与 pooling 之后添加了 1x1 卷积用来降维，从而获得了 Inception 模块的最终形态。



图片来源：https://arxiv.org/abs/1409.4842

这里有个额外的小知识点，如果是面试，经常会被问到为什么采用 1x1 的卷积或者 1x1 卷积的作用。1x1 卷积的作用就是用来升维或者降维的。

GooLeNet 就是由以上的 Inception 模块构成的一个 22 层网络。别看网络层数有 22 层，但是它参数量却比 AlexNet 与 VGG 都要少，这带来的优势就是，搭建起来的模型就很小，占的存储空间也小。具体的网络结构你可以参考它的论文。

ResNet

ResNet 中文意思是残差神经网络。在 2015 年的 ImageNet 比赛中，模型的分类能力首次超越人眼，1000 类图片 top-5 的错误率降低到 3.57%。

在论文中作者给出了 18 层、34 层、50 层、101 层与 152 层的 ResNet。101 层的与 152 层的残差神经网络效果最好，但是受硬件设备以及推断时间的限制，50 层的残差神经网络在实际项目中更为常用。

具体的网络结构你感兴趣的话可以自己看看论文全文，这里我着重带你看看这个网络的主要突破点。

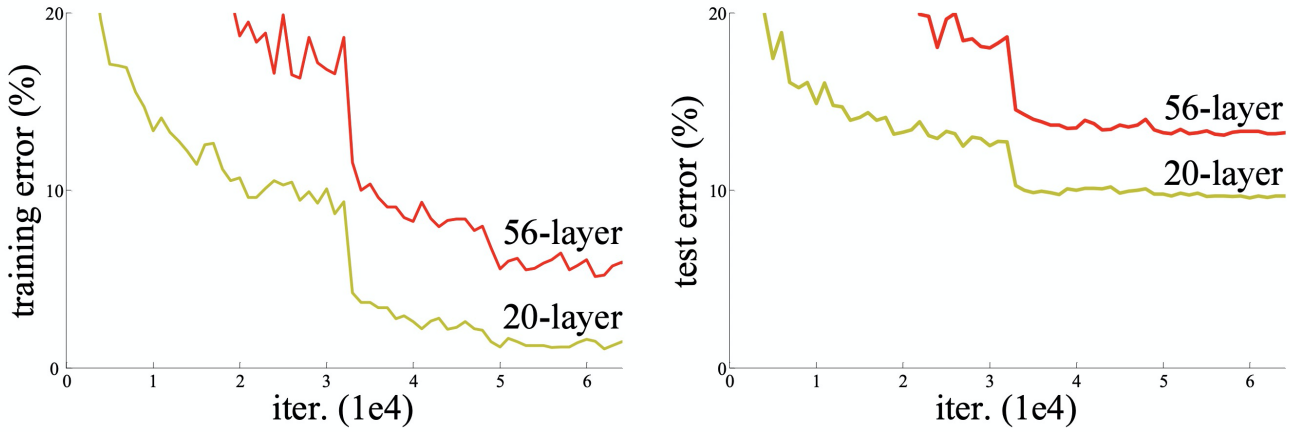
网络退化问题

虽说研究已经证明，随着网络深度的不断增加，网络的整体性能也会提升。如果只是单纯的增加网络，就会引起以下两个问题：第一，模型容易过拟合；第二，产生梯度消失、梯

度爆炸的问题。

虽然随着研究的不断发展，以上两个问题都可以被解决掉，但是 ResNet 网络的作者发现，以上两个问题被规避之后，简单的堆叠卷积层，依然不能获得很好的效果。

为了验证刚才的观点，作者做了这样的一个实验。通过搭建一个普通的 20 层卷积神经网络与一个 56 层的卷积神经网络，在 CIFAR-10 数据集上进行了验证。无论训练集误差还是测试集误差，56 层的网络均高于 20 层的网络。下图来源于论文。



图片来源：https://arxiv.org/abs/1512.03385

出现这样的情况，作者认为这是网络退化造成的。

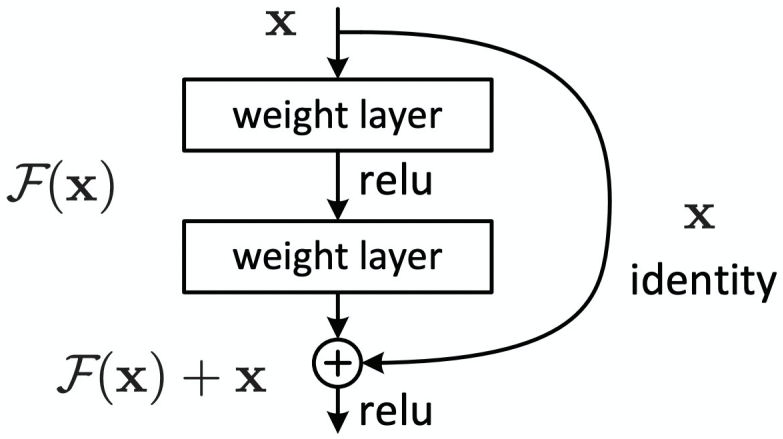
网络退化是指当一个网络可以开始收敛时，随着网络层数的增加，网络的精度逐渐达到饱和，并且会迅速降低。这里精度降低的原因并不是过拟合造成的，因为如果是过拟合，上图中 56 层的在训练集上的精度应该高于 20 层的精度。

作者认为这一现象并不合理，假设 20 层是一个最优的网络，通过加深到 56 层之后，理论上后面的 36 层是可以通过学习到一个恒等映射的，也就是说理论上不会学习到一个比 26 层还差的网络。所以，作者猜测网络不能很容易地学习到恒等映射 (恒等映射就是 $f(x)=x$)。

残差学习

正如刚才所说，从网络退化问题中可以发现，通过简单堆叠卷积层似乎很难学会到恒等映射。为了改善网络退化问题，论文作者何凯明提出了一种深度残差学习的框架。

因为网络不容易学习到恒等映射，所以就让它强制添加一个恒等映射，如下图所示（下图来源于论文）。



图片来源：https://arxiv.org/abs/1512.03385

具体实现是通过一种叫做 shortcut connection 的机制来完成的。在残差神经网络中 shortcut connection 就是恒等变换，就是上图中带有 x identity 的那条曲线，包含 shortcut connection 的几层网络我们称之为残差块。

残差块被定义为如下形式：

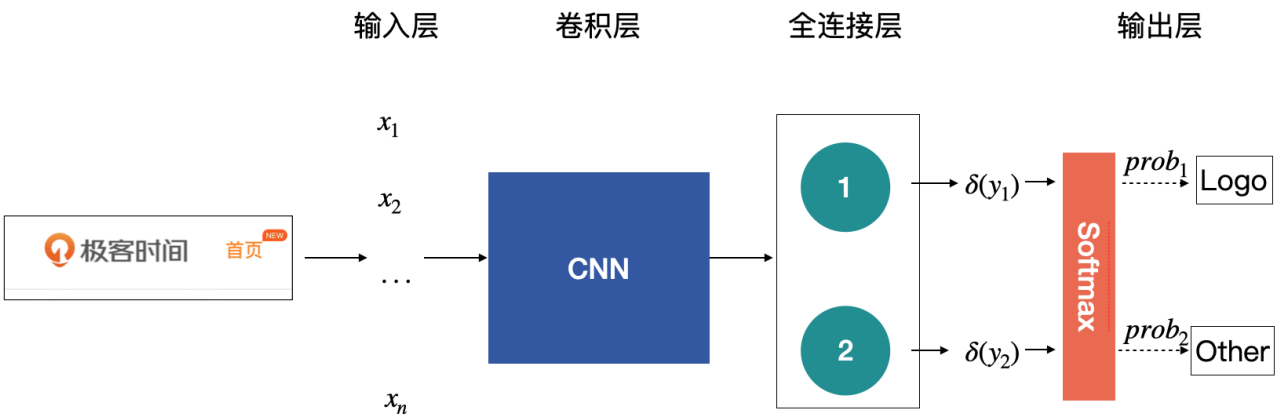
$$y = F(x, W_i) + x$$

F 可以是 2 层的卷积层。也可以是 3 层的卷积层。最后作者发现，通过残差块，就可以训练出更深、更加优秀的卷积神经网络了。

小结

恭喜你完成了这节课的学习，让我们回顾一下这节课的主要内容。

首先我们从多层感知机说起，带你认识了这个卷积神经网络的前身。之后我们一起推导出了图像分类原理的基础模型。**你需要注意的是，整个模型或者网络的重点全都在卷积神经网络那块，所以这也是我们的工作重点。**



之后我们结合业界标杆 ImageNet 的评选情况，一起学习了一些经典的网络结构：VGG、GoogLeNet、ResNet。这里为了让你快速抓住重点，我是从每个网络解决了什么问题，各自有什么突破点展开的。也建议你课余时间多读读相关论文，做更为详细深入的了解。

纵观网络结构的发展，我们不难发现，一直都是长江后浪推前浪，一代更比一代强。掌握了这些网络结构，你就是深度学习未来的弄潮儿。下节课我们再一起实践一个图像分类项目，加深你对图像分类的理解，敬请期待。

思考题

欢迎推荐一下近几年来，你自己觉得比较不错的神经网络模型。

欢迎你在留言区跟我交流互动，也推荐你把这节课分享给更多的同事、朋友。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 16 | 分布式训练：如何加速你的模型训练？

更多学习推荐

2021 最新大厂
Go 工程师面试真题

大厂面试真题 + 金九银十全新整理 + 核心知识全覆盖

免费领取

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。